Objectives:

1. Implementing the two classic and popular frequent pattern mining algorithms
2. Comparing the performance of the algorithms on various datasets with different parameter settings

Description:

Write a python/C/C++/Java code in a modular* approach that takes input from the console. The console input may contain a set of switches as parameters followed by the program name and their corresponding parameter values. Let's name the program "FPM" which stands for "Frequent Pattern Mining". Your program should be able to handle at least the following switches:

Parameters/Switches:

1. -a algoName (algoName can be "Ap" for apriori and "FP" for FP-growth. By default, Apriori should be used.)
2. -d datasetName (datasetName is the name of the dataset. Default dataset is "toy.txt" which is a text file containing the dataset given in the book as the example dataset for Apriori and Fp-Growth. The datasets are Mushroom, Kosarak, Chess, and Retail found in https://archive.ics.uci.edu/ml/datasets.php and the toy dataset found in Han's book.)
3. -t threshold (a threshold is a floating-point number stating the support threshold to be used as an input to the frequent pattern mining algorithms. By default this is 0.5 which means 50%.)
4. -m (this switch requests your program to store memory Consumption, i.e., the amount of memory used in MB by a particular run of an algorithm with the given parameter setting, i.e. threshold, dataset and algoname.)
5. -rt (this switch request your program to store runTime, i.e., the amount of time required in millisecond by a particular run of an algorithm with the given parameter setting, i.e. threshold, dataset and algoname.)
6. -o outputFileName (outputFileName represents the CSV file name where the threshold will be written in the 1st column, and the time required and memory usage will be written in the next two columns. Note, if this switch is not specified explicitly then by default the same file will be used to append for the same algorithm and same dataset. Otherwise, a new file has to be created where each file should have the following name pattern the values for switch "-a" plus an underscore "_" then the value for the switch "-d" and ".csv".)
7. -n (this switch request your program to store the num of Patterns Generated by the algorithm for this particular run.)
8. -pc patternConsole (patternConsole means the generated patterns will be printed on the console. The printing should be done in lexicographical order of the patterns.)
9. -pf patternFileName (patternFileName stands for the name of the file where the generated patterns will be written. By default or if the name is not specified or the switch is not used the file name will be the values for switch "-a" plus an underscore "_" then the value for the switch "-d" plus an underscore "_" the value for the switch "-t" if not specified explicitly by the switch "-pf" and ".txt". The writing should be done in lexicographical order of the patterns.)

NB:

1. A hidden challenge in this work is to work with real-life data and real-time performance analysis and comparison. The running time for a certain dataset for a particular threshold

should be too high that may not end in years. On the other hand for threshold on a particular dataset, there may not be any patterns available. So, choosing the right threshold for each dataset is a challenge.

2. *In order to save your time from waiting for outputs, you can write a batch file for windows or a shell script for Linux/mac where all the runs with required parameters or switches are listed. Once you run the script or batch from a command line, you may set out for other works leaving your computer dealing with the computation by itself. Just make sure no one turns your computer off while the batch/script is running. For your reference, a sample batch/script file is shown below.*
3. *Your program must include all kinds of exception handling.*
4. *In case of an invalid switch or invalid value for a particular switch, your program should show a helpful hint instead of treating the value in an inappropriate way or terminating abruptly.*
5. *At least 5 threshold values for each dataset where at least 100 patterns get generated*
6. *Output pattern files, CSV files and plots (Threshold in the X-axis vs running time in Y-axis, and Threshold in the X-axis vs memory usage in Y-axis) for the following parameters.*

Sample Batch file content with potential inputs:

```
Python FPM.py -a FP -d mushroom.txt -t 0.3 -rt -m
Python FPM.py -a AP -d mushroom.txt -t 0.3 -rt -m
```

.sh => for linux and mac
.bat => for windows